CS 2740 Knowledge representation
Lecture 19

Bayesian belief networks. Inference.

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Project proposals

Due: **Tuesday, 27, 2007**

- **1-2 pages long**

Proposal

- **Written proposal:**
  1. Outline the problem you would like to tackle. Why is the problem important?
  2. Methods you plan to try and implement for the problem. References to previous work.
  3. How do you plan to test, run your solution.
  4. Schedule of work (approximate timeline of work)
- **A 3-slide PPT presentation summarizing points 1-4**

OpenCyc installation on linux

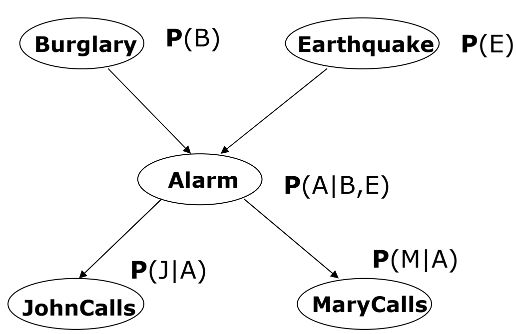
1. Note that OpenCyc for linux works with only some of our linux machines: arsenic and antimony were tested and work. So if you can't run it, try ssh arsenic.cs.pitt.edu or ssh antimony.cs.pitt.edu
2. Do **tar -xvzf ~cmason/opencyc-1.0.2-linux.tgz** from home directory.
3. Study the README.txt

- 3.1 Create a file named platform-override.txt under your opencyc-1.0/scripts directory. Type in "RH-ES3-x86_32" to your file.
- 3.2 Run ./run-cyc.sh from your opencyc-1.0/scripts directory.

This should be able to build the OpenCyc system.

Bayesian belief network.

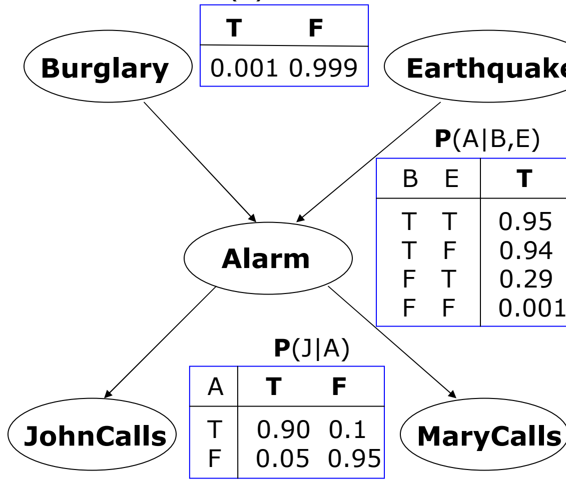
1. **Directed acyclic graph**
   - **Nodes** = random variables
   - **Links** = missing links encode independences.



Bayesian belief network

2. Local conditional distributions

- relate variables and their parents



P(B)

T	**F**
0.001	0.999

P(E)

T	**F**
0.002	0.998

P(A|B,E)

B	E	**T**	**F**
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

P(J|A)

A	**T**	**F**
T	0.90	0.1
F	0.05	0.95

P(M|A)

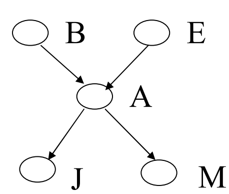
A	T	F
T	0.7	0.3
F	0.01	0.99

Full joint distribution in BBNs

Full joint distribution is defined in terms of local conditional distributions (obtained via the chain rule):

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

Example:



Assume the following assignment of values to random variables

$B=T, E=T, A=T, J=T, M=F$

Then its probability is:

$$P(B=T, E=T, A=T, J=T, M=F) =$$
$$P(B=T)P(E=T)P(A=T \mid B=T, E=T)P(J=T \mid A=T)P(M=F \mid A=T)$$

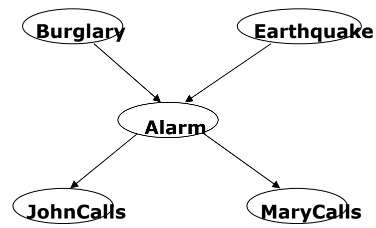
Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- **What did we save?**

Alarm example: 5 binary (True, False) variables



# of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

$$2^5 - 1 = 31$$

# of parameters of the BBN:

$$2^3 + 2(2^2) + 2(2) = 20$$

One parameter in every conditional is for free:

$$2^2 + 2(2) + 2(1) = 10$$

Inference in Bayesian networks

- BBN models compactly the full joint distribution by taking advantage of existing independences between variables
  - Smaller number of parameters
- But we are interested in solving various **inference tasks**:
  - **Diagnostic task. (from effect to cause)**

    $$\mathbf{P}(Burglary \mid JohnCalls = T)$$

  - **Prediction task. (from cause to effect)**

    $$\mathbf{P}(JohnCalls \mid Burglary = T)$$

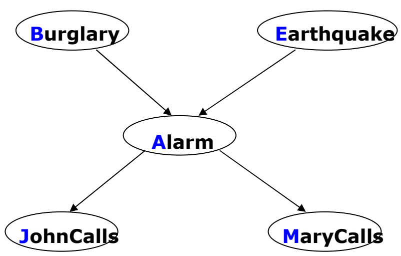
  - **Other probabilistic queries** (queries on joint distributions).

    $$\mathbf{P}(Alarm)$$

- **Question:** Can we take advantage of independences to construct special algorithms and speedup the inference?

Inference in Bayesian network

- **Bad news:**
  - Exact inference problem in BBNs is NP-hard (Cooper)
  - Approximate inference is NP-hard (Dagum, Luby)
- **But** very often we can achieve significant improvements
- Assume our Alarm network



- Assume we want to compute: $P(J = T)$

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J = T) =$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(B = b, E = e, A = a, J = T, M = m)$$

$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(J = T \mid A = a) P(M = m \mid A = a) P(A = a \mid B = b, E = e) P(B = b) P(E = e)$$

Computational cost:

Number of additions: 15

Number of products: 16*4=64

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$=\sum_{b\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)[\sum_{m\in T,F}P(M=m\mid A=a)][\sum_{b\in T,F}P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]]$$

Computational cost:

Number of additions: 1+2*[1+1+2*1]=**9**

Number of products: 2*[2+2*(1+2*1)]=16

Variable elimination

- **Idea:** interleave sum and products one variable at the time during the inference
  - Typically relies on a special structure (called **joint tree**) that groups together multiple variables
  - E.g. Query $P(J=T)$ requires to eliminate A,B,E,M and this can be done in different order

$$P(J=T)=$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

Variable elimination

Assume order: M, E, B,A to calculate $P(J=T)$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F} P(J=T\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]$$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F} P(J=T\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e) \quad 1$$

$$= \sum_{a\in T,F}\sum_{b\in T,F} P(J=T\mid A=a)P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]$$

$$= \sum_{a\in T,F}\sum_{b\in T,F} P(J=T\mid A=a)P(B=b)\,\tau_1(A=a,B=b)$$

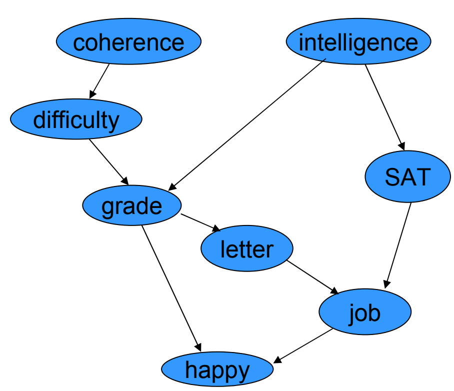
$$= \sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{e\in T,F} P(B=b)\,\tau_1(A=a,B=b)\right]$$

$$= \sum_{a\in T,F} P(J=T\mid A=a)\quad \tau_2(A=a)$$

Variable elimination

The order in which variables are eliminated may effect the efficiency of the variable elimination process

Assume the following BBN and calculation of *P(Job)*:



Variable elimination

Calculations performed in terms of factors:

$$p(J) = \sum_{L,S,G,H,I,D,C} \phi(c)\phi(i)\phi(d,c)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j)$$

$$= \sum_{L,S,G,H,I,D} \phi(i)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j)\sum_{C}\phi(c)\phi(d,c)$$

$$= \sum_{L,S,G,H,I,D} \phi(i)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j)\tau(d)$$

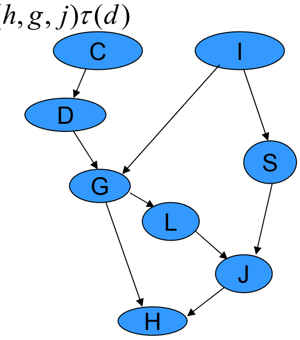
$$\ldots$$

$$= \sum_{L,S}\phi(j,l,s)\sum_{G}\phi(l,g)\tau(s,g)\tau(g,j)$$

$$= \sum_{L,S}\phi(j,l,s)\tau(l,s,j)$$

$$= \sum_{L}\tau(l,j)$$

$$= \tau(j)$$



Factor Product

Variables: A,B,C

$$\phi(A,B,C) = \phi(A,C) \circ \phi(A,B)$$

$\phi(A,C)$

b1	c1	0.1
b1	c2	0.6
b2	c1	0.3
b2	c2	0.4

$\phi(A,B)$

a1	b1	0.5
a1	b2	0.2
a2	b1	0.1
a2	b2	0.3
a3	b1	0.2
a3	b2	0.4

$\phi(A,B,C)$

a1	b1	c1	0.5*0.1
a1	b1	c2	0.5*0.6
a1	b2	c1	0.2*0.3
a1	b2	c2	0.2*0.4
a2	b1	c1	0.1*0.1
a2	b1	c2	0.1*0.6
a2	b2	c1	0.3*0.3
a2	b2	c2	0.3*0.4
a3	b1	c1	0.2*0.1
a3	b1	c2	0.2*0.6
a3	b2	c1	0.4*0.3
a3	b2	c2	0.4*0.4

Factor Marginalization

Variables: A,B,C

$$\phi(A,C) = \sum_{B} \phi(A,B,C)$$

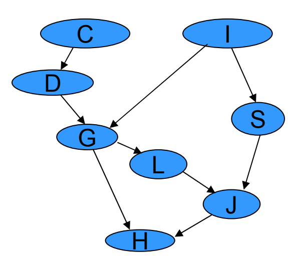
a1	b1	c1	0.2
a1	b1	c2	0.35
a1	b2	c1	0.4
a1	b2	c2	0.15
a2	b1	c1	0.5
a2	b1	c2	0.1
a2	b2	c1	0.3
a2	b2	c2	0.2
a3	b1	c1	0.25
a3	b1	c2	0.45
a3	b2	c1	0.15
a3	b2	c2	0.25

a1	c1	0.2+0.4=0.6
a1	c2	0.35+0.15=0.5
a2	c1	0.8
a2	c2	0.3
a3	c1	0.4
a3	c2	0.7

Variable elimination

Trace 1:

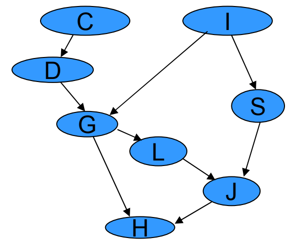
Step	Var	Factors Used	New Factor
1	C	$\phi_c(C), \phi_D(D,C)$	$\tau_1(D)$
2	D	$\phi_G(G,I,D), \tau_1(D)$	$\tau_2(G,I)$
3	I	$\phi_I(I), \phi_S(S,I), \tau_2(G,I)$	$\tau_3(G,S)$
4	H	$\phi_H(H,G,J)$	$\tau_4(G,J)$
5	G	$\tau_4(G,J), \tau_3(G,S), \phi_L(L,G)$	$\tau_5(J,L,S)$
6	S	$\tau_5(J,L,S), \phi_J(J,L,S)$	$\tau_6(J,L)$
7	L	$\tau_6(J,L)$	$\tau_7(J)$



Variable elimination

Trace 1:

Step	Var	Factors Used	New Factor
1	C	$\phi_c(C), \phi_D(D, C)$	$\tau_1(D)$
2	D	$\phi_G(G, I, D), \tau_1(D)$	$\tau_2(G, I)$
3	I	$\phi_I(I), \phi_S(S, I), \tau_2(G, I)$	$\tau_3(G, S)$
4	H	$\phi_H(H, G, J)$	$\tau_4(G, J)$
5	G	$\tau_4(G, J), \tau_3(G, S), \phi_L(L, G)$	$\tau_5(J, L, S)$
6	S	$\tau_5(J, L, S), \phi_J(J, L, S)$	$\tau_6(J, L)$
7	L	$\tau_6(J, L)$	$\tau_7(J)$

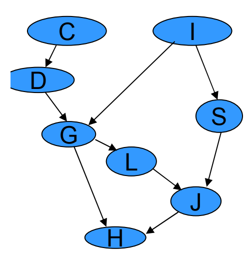


Complexity: 4 variables used – 1 summed away

Variable elimination

Trace 2:

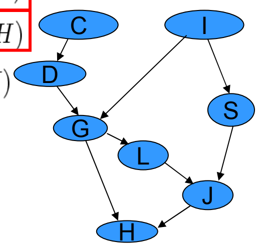
Step	Var	Factors Used	New Factor
1	G	$\phi_G(G, I, D), \phi_L(L, G)\phi_H(H, G, J)$	$\tau_1(I, D, L, J, H)$
2	I	$\phi_I(I), \phi_S(S, I)\tau_1(I, D, L, J, H)$	$\tau_2(D, L, S, J, H)$
3	S	$\phi_J(J, L, S), \tau_2(D, L, S, J, H)$	$\tau_3(D, L, J, H)$
4	L	$\tau_3(D, L, J, H)$	$\tau_4(D, J, H)$
5	H	$\tau_4(D, J, H)$	$\tau_5(D, J)$
6	C	$\tau_5(D, J), \phi_D(D, C)$	$\tau_6(D, J)$
7	D	$\tau_6(D, J)$	$\tau_7(J)$



Variable elimination

Trace 2:

Step	Var	Factors Used	New Factor
1	G	$\phi_G(G,I,D), \phi_L(L,G)\phi_H(H,G,J)$	$\tau_1(I,D,L,J,H)$
2	I	$\phi_I(I), \phi_S(S,I)\tau_1(I,D,L,J,H)$	$\tau_2(D,L,S,J,H)$
3	S	$\phi_J(J,L,S), \tau_2(D,L,S,J,H)$	$\tau_3(D,L,J,H)$
4	L	$\tau_3(D,L,J,H)$	$\tau_4(D,J,H)$
5	H	$\tau_4(D,J,H)$	$\tau_5(D,J)$
6	C	$\tau_5(D,J), \phi_D(D,C)$	$\tau_6(D,J)$
7	D	$\tau_6(D,J)$	$\tau_7(J)$



Complexity: 6 variables used – 1 summed out

Inference in Bayesian network

- **Exact inference algorithms:**
  - **Variable elimination**
  - Recursive decomposition (Cooper, Darwiche)
  - Belief propagation algorithm (Pearl)
  - Arc reversal (Olmsted, Schachter)

- **Approximate inference algorithms:**
  - **Monte Carlo methods:**
    - Forward sampling, Likelihood sampling
  - Variational methods